DynTok: Dynamic Compression of Visual Tokens for Efficient and Effective Video Understanding

Anonymous ACL submission

Abstract

Typical video modeling methods, such as LLava, represent videos as visual tokens, which are then processed by the LLM backbone for effective video understanding. However, this approach would result in a massive number of visual tokens especially on long videos. A practical approach is to first extract relevant visual information from the large visual context before integrating it into the LLM backbone, thereby avoiding substantial computational overhead. In this work, we propose DynTok, a training-free strategy for **Dyn**amic video **Tok**en compression. DynTok adaptively splits visual tokens into groups and merges them within each group, achieving high compression in low-information-density areas, while retaining essential information. Our method reduces the token count to 44.4% of the original size while maintaining comparable performance. It benefits more from increasing the number of video frames and achieves 65.3% on Video-MME and 72.5% on MLVU. By leveraging this simple yet effective compression method, we uncover the redundancy in video token representations and provide insights into designing more efficient video modeling techniques.

1 Introduction

Multimodal understanding models are rapidly advancing, showcasing remarkable capabilities in image and video understanding. Typically, Vision-Language Models (VLMs) (Bordes et al., 2024) leverage CLIP to extract visual representations, which are then processed through a connector and finally fed into the large language model (LLM) backbone. Compared to image-understanding tasks, video understanding involves processing a large number of video frames. Efficient encoding of the numerous video frames is the core topic of video understanding. For instance, a 10-minute video sampled at 1 frame per second (fps) generates 600 frames. If each frame is modeled with 196 tokens, this results in 117,600 visual tokens being fed into the LLM. For hour-long videos, this can escalate to over 700k context tokens, making comprehension of such long-duration videos exceptionally challenging. This creates significant obstacles to effectiveness and efficiency. On one hand, extracting relevant information from such extensive contexts is inherently complex. On the other hand, handling ultra-long contexts demands substantial computational resources and memory capacity.

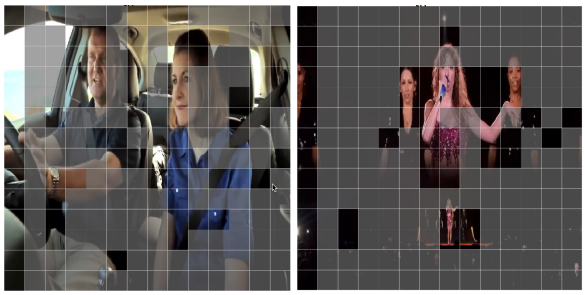

Figure 1: Illustration of two test video frames with their visual token grouping results by DynTok. Each image patch is determined to be masked or not based on its patch similarity to its left neighbor. In this figure, for each patch, if the similarity is greater than the threshold 0.6, it is masked with a gray block. As can be seen, DynTok can effectively retain informative patches with minimal information loss, making it suitable for integration into LLMs for diverse video understanding tasks.

A number of studies have focused on reducing the number of tokens input into LLMs to fit within the context window and improve computational efficiency. For example, LLaVA (Zhang et al., 2025) and Dynamic-VLM (Wang et al., 2024a) perform pooling operations on visual tokens, while Q-former (Li et al., 2023a) compresses visual tokens into a fixed-length token sequence (Bai et al., 2023; Xiao et al., 2021). Some works also select important visual tokens based on attention weights in ViT (Yang et al., 2024b) or fuse tokens based

on similarities (Tao et al., 2024), inspired by Bolya et al. (2023). However, pooling-based methods often fail to account for the varying importance of different tokens, typically using a 2x2 pooling size, with larger pooling sizes leading to performance degradation. Furthermore, current similarity-based token fusion, which is applied across the entire image or video, struggles to preserve spatial information and incurs high computational costs for similarity calculations. In this work, we explore a local token reduction method that more effectively preserves spatial information while minimizing computational overhead.

The main idea of DynTok is illustrated in Figure 1. A video frame is typically processed into multiple visual tensors via pretrained visual encoders such as CLIP/SigLIP (Zhai et al., 2023), each corresponding to image patches (e.g., 14x14 in LLaVA-OneVision (Li et al., 2025)). It could be observed that the informativeness of these patches varies. For example, in the right image, compared to its main subject, the dark background area carries less informational content, indicating the greater potential for token compression. Typically, solid color-filled low-density areas exhibit consecutive similar patches, while high information-density regions such as the edges of people or objects differ from preceding tokens. Leveraging this phenomenon, we propose to adaptively compress less informative patch tokens, thereby reducing the proportion of low-information-density tokens and decreasing the total number of visual tokens. Furthermore, by merging highly similar tokens within the same row, our method better preserves the spatial relationships of visual tokens while maintaining computational efficiency.

We validated DynTok through extensive experiments on multiple video benchmark datasets. DynTok achieved a slight improvement in model performance while compressing the total number of video tokens to 44.4% of the baseline method, corresponding to a 2.2x reduction. Moreover, thanks to the compression of tokens per frame, DynTok demonstrates stronger robustness to more extracted frames. In long video tasks, where more frames are processed, DynTok demonstrates further performance improvements, achieving a VideoMME accuracy of 65.3%. These results highlight DynTok's dual benefits: it not only improves modeling efficiency but also reduces the complexity for LLMs to extract information from lengthy sequences of visual tokens. Our main contribution could be summarized as follows:

- We propose a simple yet effective token compression method, DynTok. It achieves high compression ratios in low-information-density patches while effectively preserving critical information by adaptively splitting and fussing the similar adjacent tokens.
- Experiment demonstrates that DynTok reduces video tokens to 44.4% of the baseline (a 2.2x reduction) while maintaining performance. DynTok could benefit more from the increasing of video frames, enhancing long video understanding by 1.5% on MLVU and 1.7% on Video-MME while using a similar number of tokens compared to the baseline.

2 Related works

2.1 Visual language model for video understanding

VLMs have achieved significant milestones in both image-text integration (OpenAI, 2023, 2024; Liu et al., 2023; Alayrac et al., 2022; Tong et al., 2024; Li et al., 2023a) and video comprehension (Team, 2024; Li et al., 2025; Zhang et al., 2024b; Wang et al., 2024b). Unlike image understanding, video modeling requires processing sequences of images and capturing the dynamic relationships between frames. Recent works, such as MiraData (Ju et al., 2025) and LLaVA-Video (Zhang et al., 2024b), focus on leveraging existing VLMs to generate high-quality video captioning or question-answering datasets. Meanwhile, models like VideoChat (Li et al., 2023b), VideoLLaMa2 (Cheng et al., 2024), and Aria (Li et al., 2024a) explore video encoding modules, methods for integrating video encoders with large language models (LLMs), and modifications to LLM architectures. Additionally, there has been significant work on handling long video contexts (Weng et al., 2024; Xue et al., 2024).

However, many of these approaches rely on fixed token sizes for distinct frames to represent video, leading to redundant and lengthy video inputs. This paper addresses the challenge of reducing redundancy in video data through spatial compression, aiming to mitigate model overload during video processing.

2.2 Visual token compression

Video inputs are typically represented as separate frames, resulting in a substantial increase in the

number of visual tokens as the video lengthens. Existing video token compression methods can be broadly classified into two categories: strategy-based and learning-based approaches.

Strategy-based methods focus on dynamically resizing or merging visual tokens. For example, LLaVA-Video (Zhang et al., 2024c) divides frames into high-resolution parts for detailed information and low-resolution parts for dynamic content. Dynamic-VLM compresses similar visual tokens within each frame, while FrameFusion (Fu et al., 2024b) employs temporal merging and spatial pruning in a two-stage compression process. Additionally, methods like LLaVA-PruMerge (Shang et al., 2024) and VisionZip (Yang et al., 2024b) leverage attention distributions to guide token selection. While these approaches effectively reduce token redundancy, they may struggle to accurately capture the spatial and temporal distribution of visual tokens, potentially compromising the model's ability to understand the full context of the video.

Learning-based methods, in contrast, train compressors to directly compress the tokens corresponding to each frame, learning optimal compression patterns through model training. For instance, MobileVLM (Chu et al., 2023), TokenPacker (Li et al., 2024c), MQT (Hu et al., 2024), and MiniCPM-V (Yao et al., 2024) reduce the number of visual tokens per frame using purpose-built downsampling projectors. Blip3-Video (Ryoo et al., 2024) and InternVideo2 (Wang et al., 2024c) employ temporal encoder layers to aggregate and compress visual tokens over time. Similarly, VideoChat (Li et al., 2023b) and LLaVA-mini (Zhang et al., 2025) use multi-stage fusion strategies to further compress video inputs. However, many of these methods apply uniform compression, reducing each frame to a fixed number of tokens. These approaches overlook the dynamic nature of visual redundancy, where the amount of compression needed can vary across frames based on content complexity and relevance. As a result, these methods may not fully exploit the potential for more efficient token compression in videos.

3 Method

3.1 Preliminary

Given one or more videos, denoted as V, and a question q, the goal of video understanding is to generate the corresponding answer a based on the content of the video. We adopt the LLaVA architecture (Li et al., 2025), which effectively integrates the pretrained LLM with visual inputs. The architecture consists of three main components: a vision encoder, an LLM backbone, and an MLP connector. The vision encoder processes video frames to extract visual representations. These representations are then mapped to visual tokens in the word embedding space of the LLM through the MLP connector. Finally, the LLM leverages the visual information to fully comprehend the query and generate a response. As mentioned in the introduction, the token sequence for video understanding could be very long and the computational complexity of an LLM can be quadratic with respect to the sequence length. Moreover, the visual tokens account for the majority. DynTok processed the visual tokens before they were fed into the LLM.

Formally, suppose t frames are extracted from the input video. The visual encoder encodes the frames into a tensor X with shape (t, h, w, d_{clip}), where h and w are the number of patches in the horizontal and vertical directions respectively, and d_{clip} is the representation dimension. Then the MLP layer transforms the representation into the corresponding visual tokens $H \in \mathbb{R}^{t,h,w,d_{emb}}$, where d_{emb} is the dimension of the LLM embedding layer.

3.2 Token level compression

Typically, H will be flatten into the visual token tensor with size $(t \times h \times w, d_{emb})$ and then fed into the LLM. Instead of adopting all tokens, DynTok compresses the visual tokens into $H' \in \mathbb{R}^{l,d_{emb}}$ based on the visual representation X, where $l < t \times h \times w$ is the number of reduced token.

For each video frame represented into a tensor of size (h, w, d_{emb}), i.e. there are (h, w) tensors corresponding to (h, w) image patches, the main idea of DynTok is to merge adjacent visual tokens within each row. We first describe the specific method for calculating token similarity, followed by the process of dynamically constructing token groups and implementing dynamic information compression.

Token similarity measurement Each image patch is represented as $X_{i,j,k} \in \mathbf{R}^{d_{clip}}$, and then transformed into a visual token $H_{i,j,k} \in \mathbf{R}^{d_{emb}}$. We simplify the notation to $x_k \in \mathbf{R}^{d_{clip}}$ and $h_k \in \mathbf{R}^{d_{emb}}$, by omitting the first two dimensions here. The similarity between two adjacent visual tokens is computed based on the CLIP representation

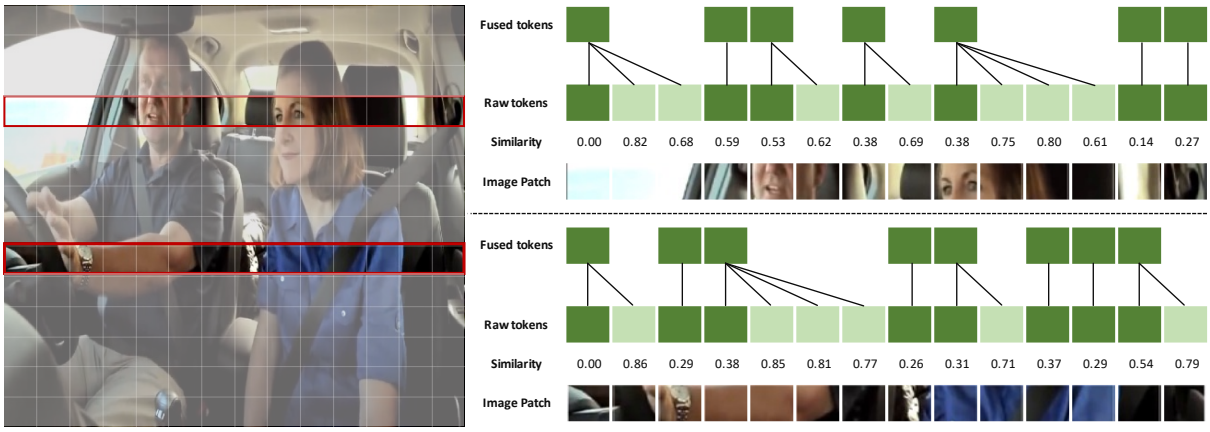

Figure 2: Illustration of the proposed token compression method, DynTok. For each row of image patches, every visual token of a patch is compared with the one corresponding to its left patch. If the similarity exceeds a predefined threshold (0.6 in this example), the token merges into the preceding group; otherwise, it initiates a new group as a primary token. Tokens in the same group are fused together to reduce the number of visual tokens finally.

tensor:

$$s_{(k-1,k)} = \frac{x_{k-1} \cdot x_k}{\|x_{k-1}\|\|x_k\|}.$$

The CLIP representation is used here, by considering that the CLIP or SigLIP trained with contrastive learning cosine loss works better with cosine similarity than with the visual tokens in the embedding space (Zhou et al., 2022).

Dynamic token merging It is interesting to observe that adjacent image patches with solid-color blocks, typically have a high similarity with adjacent nodes in Figure 1. DynTok aims to fuse the adjacent image patches to reduce the number of visual tokens while keep the most information.

For each row of image patches $(h_0, h_1, ..., h_{w-1}) \in \mathbb{R}^{(w,d_{emb})}$, these patches are split into groups based on a similarity threshold hyperparameter S_{th}. Initially, the first token patch forms a new token group. For each subsequent visual token h_k, if the similarity $s_{(k-1,k)}$ between this token and its predecessor exceeds the threshold, it indicates the token carries little new information. In this case, h_k is added to the token group containing h_{k-1}. If the similarity does not surpass the threshold S_{th}, it suggests a significant content difference, and thus a new token group is created starting from h_k. The visual token groups could be formally denoted as $[(h_s, h_{s+1}, .., h_e)_{h'}]$, and we denote the number of groups as h'. $h' \leq h$ is a dynamic value depending on the similarity between adjacent tokens.

At the fusing stage, the visual tokens within each token group are averaged to form a new visual token tensor of size (l', d_{emb}). However, the number of tokens at each row varies, making it difficult for the LLM to determine the spatial information of each token, e.g. LLM is not informed where a new row starts. Therefore, we add a grid marker at the end of each row to keep the spatial information. Finally, the visual tokens from different rows are concatenated together as $H' \in \mathbb{R}^{l,d_{emb}}$.

In contrast to static pooling methods like bilinear pooling (Wang et al., 2024a), we dynamically construct token groups based on token similarity. By fusing tokens with low information density, we can maintain a good modeling performance with fewer tokens. DynTok, as a local token merge strategy, can better preserve spatial information, and it only requires calculating the similarity between the current token and its preceding token. The computational complexity has a linear relationship with the token length.

3.3 Model Training

DynTok is a parameter-free token compression method that merges visual tokens carrying similar information. As a result, DynTok can be applied in a zero-shot manner to existing models. However, using in the zero-shot manner may encounter a training-inference mismatch issue. To mitigate this, the model can be trained according to the DynTok configuration to become familiar with the compressed tokens.

4 Experiments

4.1 Training Configurations and Evaluations

Training data. The training data is divided into two parts: single-image and video data. Firstly, we

Model	MVbench	PercepTest	NextQA	LongVideo	MLVU	VideoMME
Proprietary models						
GPT4o (OpenAI, 2024)	-	-	-	66.7	64.6	71.9
Gemini1.5-pro (Team, 2024)	-	57.5	-	64.0	-	75.0
Claude3.5-sonnet (Anthropic, 2024)	-	-	-	-	-	60.0
GPT4V (OpenAI, 2023)	43.7	-	-	61.3	49.2	59.9
GPT4o-mini (OpenAI, 2024)	-	-	-	-	-	64.8
Open-source models						
Qwen2-VL-7B (Wang et al., 2024b)	67.0	62.3	-	-	-	63.3
Intern-VL2.5-8B (Chen et al., 2024)	72.0	-	-	68.9	60.0	64.2
LLaVA-OV-7B (Li et al., 2025)	56.7	57.1	79.4	64.7	56.5	58.2
LLaVA-Video-7B (Zhang et al., 2024b)	58.6	67.9	83.2	58.2	70.8	63.3
Dynamic-VLM (Wang et al., 2024a)	-	68.8	-	-	65.0	60.9
DyCoke-7B(Tao et al., 2024)	58.0	57.6	78.5	-	-	59.5
Models with comparable training setting						
Baseline	67.9	74.0	84.1	60.1	71.0	63.6
DynTok -$zeroshot$	65.2	74.0	83.5	59.9	70.8	62.5
DynTok	67.5	73.5	83.7	60.4	71.3	64.0
DynTok -$moreframes$	-	-	-	60.7	72.5	65.3

Table 1: Performance on various video benchmarks. Results of other models are collected from official reports of the model or the leaderboard of the benchmark. The first three datasets consist of short videos, each with a duration of less than 45 seconds, while the remaining three contain long videos ranging from several minutes to hours.

train the model with single-image data to acquire basic visual understanding, the trianing data is identical to the LLAVA-OneVision(Li et al., 2025) single-image stage. Our Video dataset consists of LLAVA-Video (Zhang et al., 2024b), VideoInstruct (Maaz et al., 2024b), VCG-Plus (Maaz et al., 2024a), and diverse video question answering/classification data from the training mixture of (Li et al., 2024b). We remove videos that could not be downloaded or opened, resulting in a final dataset of 1.79M training samples for the video stage.

Model configurations and hyper-parameters settings. We first drive a model with single-image understanding ability as the warm-up stage before our video training stage. We adopt the model architecture, training procedure and dataset following LLaVA-OneVision (Li et al., 2025), except we utilize Qwen2.5-7B-instruct (Yang et al., 2024a) as our LLM Backbone. We do not apply dynamic token compression during this stage. After that, we train the video understanding model mostly following the experimental setting of LLaVA-Video (Zhang et al., 2024b). We first resize each frame to a fixed size of 378×378 and leverage SigLIP to encode this new image, resulting with an visual matrix of 28×28. The bilinear pooling of stride 2 is performed to reduce the grid into 14×14, i.e. there are 196 tokens for each frame. And we apply DynTok on these visual tokens.

In the training, we randomly set the value S_{th} from a set $\{0.4, 0.45, 0.5, 0.55, 0.6\}$ in order to support different compression ratio. Learning rate of the MLP adapter and LLM backbone is set to 1e-5, and 2e-6 for the ViT tower. As mentioned in Section 3.2, a grid marker is added to the end of each row in order to keep the spatial information. The model is trained for 1 epoch on our dataset with the global batch-size 512. For a fair comparison, we train a baseline model using identical setting without the DynTok proposal. The experiment is conducted on 16 nodes with NVIDIA 8xH100 GPUs. With DeepSpeed Zero2 optimization, the video training stage taking approximately 16 hours.

Evaluation. We evaluate our model on six benchmarks: MVBench (Li et al., 2024b), Perception-Test (Pătrăucean et al., 2023), NextQA (Xiao et al., 2021), LongVideoBench (Wu et al., 2024), MLVU (Zhou et al., 2024) and Video-MME (Fu et al., 2024a), covering both the perception and reasoning tasks on videos of various durations. MVBench, PerceptionTest and NextQA consist of relative short videos, and the average duration are 16, 23 and 44 seconds respectively. LongVideoBench and MLVU focus on long videos, with the average video duration of 473 and 651 seconds. Video-MME is a comprehensive video

understanding benchmark and the videos are spited into short, medium and long with 0~2, 4~15 and 30~60 minutes respectively. The evaluation is performed by utilizing LMMs-Eval (Zhang et al., 2024a) with the default configuration as LLaVA-Video and LLaVA-OneVision.

4.2 Main Results

Results are listed in Table 1. First of all, our baseline model, trained on 10 million samples, achieves performance close to leading open-source models across a range of benchmarks. Building upon the collected dataset, we establish a strong comparative baseline for our model. We would make this dataset open avaliable, provide the community with a lightweight near SOTA training dataset. The baseline model typically get best performance on short video tasks with max number of frames set to 64 for the first three benchmarks with short video, and 96 frames for the long video benchmarks.

In Zero-shot DynTok, we apply the parameter-free DynTok method directly to the baseline model with S_{th} set to 0.6, and keep other settings unchanged. Without training on any fused tokens, DynTok$zero$ reduces the number of visual tokens over 2×, while preserving accuracy above 98% on PerceptionTest, NextQA, LongVideo, MLVU, and Video-MME, and 96.0% on MVbench. The result indicates that DynTok effectively keeps the most informative visual tokens. The impact on different tasks would be covered in Section 4.3.

With the integration of the DynTok strategy during model training, we achieve a more than 2× reduction in visual tokens. Unlike DynTok-$zero$ which experiences a slight drop in accuracy, DynTok surpasses the baseline model while maintaining the same number of input video frames. By consolidating similar visual tokens, the computational cost can be significantly reduced. Meanwhile, the DynTok strategy simplifies information retrieval from the long context of visual tokens, thereby even slightly enhancing model performance.

It is also observed that the DynTok model could benefit from more input frames on long video understanding compared to the baseline. The results are listed in the last row of Table 1 and more details are shown in Figure 4. DynTok achieves 65.3% accuracy on Video-MME with 160 frames extracted, resulting in 15k tokens. This is roughly equivalent to the baseline's token consumption when using 64 frames.

4.3 In-Depth Analysis

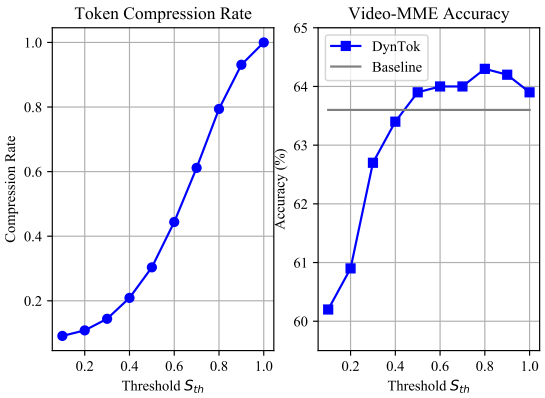

Figure 3: Results of varying the token merging threshold S_{th} on the visual token compression ratio and the Video-MME accuracy. The frame number of the baseline and DynTok is set to 96.

Different compression ratios. Figure 3 shows the visual token compression ratio and the corresponding accuracy of Video-MME under different similarity threshold S_{th}'s. As expected, a lower threshold leads to higher compression of visual tokens. With the threshold S_{th} set to 0.4, 0.5, and 0.6, the number of visual tokens is reduced to 20%, 30%, and 44.4%, respectively, corresponding to 5×, 3×, and 2.3× compression. When S_{th} is set to 0.4, despite a 5× compression, i.e. the total token number 20.1k is reduced to approximately 4k, DynTok performs comparably to the baseline (63.4% vs. 63.6%). Increasing S_{th} to 0.5 achieves a 3× compression while surpassing the baseline accuracy. Further increasing S_{th} initially leads to a slight improvement, followed by a minor decline. Overall, setting S_{th} at the range of 0.4 to 0.6 provides an optimal balance between efficiency and the model performance.

Increasing the number of frames. The model performance with different frames on the baseline and DynTok are shown in Figure 4. Overall, DynTok shows stable performance improvement as the number of frames increases. DynTok introduces a constant compression ratio on the number of visual tokens and outperforms the baseline across different numbers of extracted frames. While the baseline performance declines with 128 frames or more, DynTok continues to show improvements as the number of frames increases to 128 and 160. These results indicate that DynTok effectively preserves visual knowledge while simultaneously re-

Model	Overall	Perception tasks							Reasoning tasks				Other
		Temporal	*Spatial*	*Attribute*	*Action*	*Object*	*OCR*	*Counting*	*Temporal*	*Spatial*	*Action*	*Object*	*Synopsis*
Baseline	63.6	67.3	72.2	76.6	64.5	71.5	68.4	40.3	49.2	80.4	54.7	59.5	78.6
DynTok$zero$	62.5	*69.1*	70.4	77.9	64.2	70.1	*64.0*	38.4	49.7	80.4	*50.9*	59.5	77.4
DynTok	64.0	74.6	70.4	77.5	62.3	70.9	69.8	41.4	51.4	78.6	54.7	61.9	77.7

Table 2: Model performance across different tasks of Video-MME benchmark.

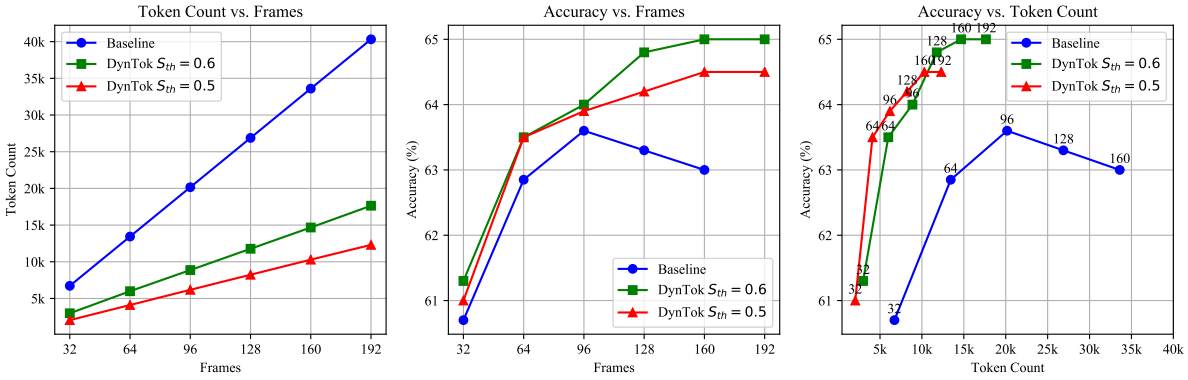

Figure 4: Model performance on Video-MME by varying numbers of video frames used.

ducing the number of tokens, thereby easing the difficulty for the LLM to extract information from long visual token sequences.

Finally, the right subplot shows the number of visual tokens and the accuracy. DynTok consistently outperforms the baseline on different visual token budgets. The results analysis also helps guide reasonable parameter configurations across different token budgets. For token budgets below 10k, setting S_{th} to 0.5 yields better performance than 0.6 with more video frames. With a higher token budget, it is preferable to increase the number of extracted frames while slightly decreasing the compression ratio. Essentially, this is a balance between increasing the number of observed frames and the compression ratio per frame.

Impact on different tasks. We analyze the impact of DynTok on different kinds of tasks on Video-MME, and the results are listed in Table 2. For DynTok$zero$, performance declines are most noticeable in OCR and action recognition/reasoning tasks. Only slight improvements are observed in temporal perception and reasoning tasks. Performance on the rest tasks remains generally stable. For OCR and action recognition tasks, directly merging tokens may lead to the confusing on fine-grained details, thereby harming the model's recognition capability. However, for temporal perception and reasoning tasks, which rely on cross-frame information fusion, reducing the token count per frame enhances the model's ability to capture cross-frame information.

After training with the DynTok strategy, performance of OCR, action recognition, and reasoning tasks either matches or surpasses the baseline. Meanwhile, the advantage in temporal perception tasks is further enhanced. Overall, the model adapts well to the token compression scenario, achieving performance improvement over the baseline with only 44.4% of the visual tokens.

Case study. We analyze the similarity between image patches and visualize the start patch of the visual token grouping in Figure 5 and Figure 6 (as well as Figure 1). The token grouping results of DynTok follow a horizontal structure, where the first visual token in each row always initiates a new group. Tokens with high consistency with preceding tokens are split into the same group on the left and shaded in gray in the figure.

Figure 5 presents the results for the same image under different threshold S_{th} values. When S_{th} is set to 0.4, most patches in each row are assigned to one group. At a threshold of 0.5, key regions in the image, such as the woman's face and the man's

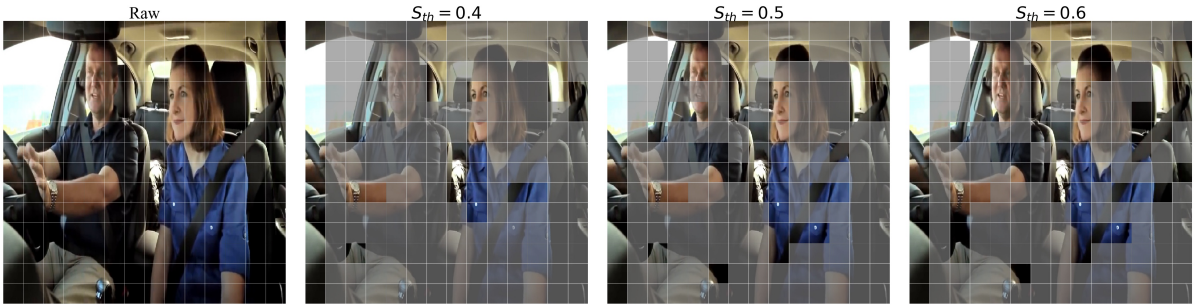

Figure 5: The impact of varying the token merging threshold on token grouping results for the same test image.

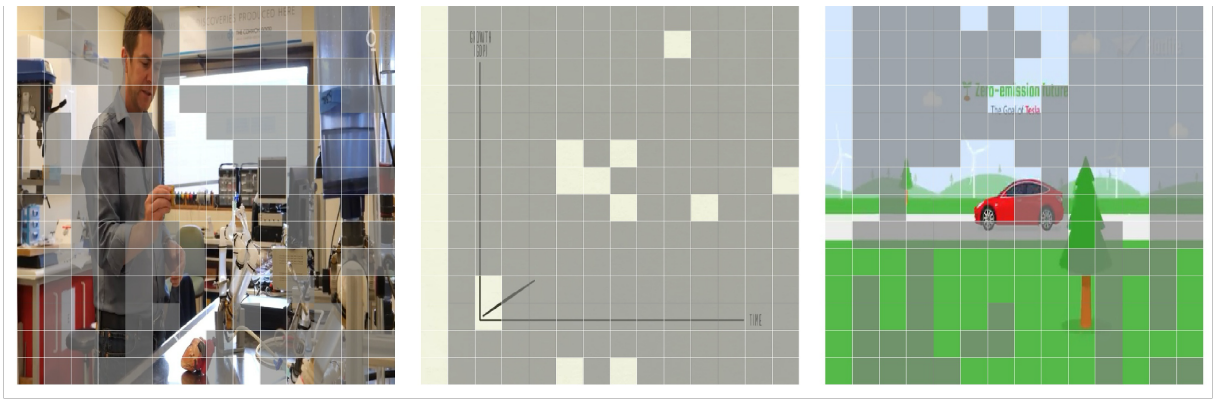

Figure 6: The token grouping results on different test images. The token merging threshold is set as 0.6.

watch, form new visual token groups. When S_{th} is raised to 0.6, important visual patches are better preserved, while background regions, such as the lathe's color and the dark area in the lower right corner, undergo greater compression.

We now investigate the results across different test cases. The black background behind the singer in the right image of Figure 1, the window in middle of the left image in Figure 6, and the solid-colored filled areas in the last two images in Figure 6 are integrated into larger token groups, leading to higher compression. In contrast, the complex scene in the left image in Figure 6 retains more tokens. This demonstrates how DynTok dynamically compresses low-information-density regions, effectively reducing the number of visual tokens. For the two rightmost cartoon video frames, ViT struggles with fine lines in the first one, causing coordinate areas to blend into the background, whereas colored regions effectively retain key elements such as the car and trees.

Overall, DynTok achieves reasonable token grouping, effectively initiating new visual token groups for high-information patches while merging tokens that carry similar information.

5 Conclusion

In this work, we introduce DynTok, a simple yet effective token compression method that dynamically merges adjacent similar tokens. DynTok significantly enhances the efficiency of video modeling while reducing the computational burden associated with processing long sequences of visual tokens. By retaining only 44% of the original visual tokens, our approach achieves performance parity with baseline models while substantially improving computational efficiency. Notably, DynTok demonstrates superior scalability with extended input video frames, achieving 65.3% accuracy on Video-MME and 72.5% accuracy on MLVU. These results highlight DynTok's ability to not only improve modeling efficiency but also to facilitate easier information extraction for LLMs from lengthy visual token sequences. Overall, DynTok offers a promising approach for designing more efficient and scalable video modeling techniques.

Limitations

In principle, this method can be applied to both images and videos. However, since the number of visual tokens in long-video modeling could be huge, we only focus on improving the efficiency

and effectiveness of video modeling, and leaves the impact on image understanding as a future work. We have not explored the coupling of this token compression method with a series of other methods, such as the scheme of gradually discarding tokens in different layers of the LLMs which is expected to further improve the efficiency.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, and Malcolm Reynolds. 2022. Flamingo: a visual language model for few-shot learning. In NeurIPS.

Anthropic. 2024. Claude-3.5. https://www.anthropic.com/news/claude-3-5-sonnet.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2023. Token merging: Your ViT but faster. In International Conference on Learning Representations.

Florian Bordes, Richard Yuanzhe Pang, Anurag Ajay, Alexander C. Li, Adrien Bardes, Suzanne Petryk, Oscar Mañas, Zhiqiu Lin ..., Jun Chen, Kushal Tirumala, Rim Assouel, Mazda Moayeri, Arjang Talattof, Kamalika Chaudhuri, Zechun Liu, Xilun Chen, Quentin Garrido, Karen Ullrich, Aishwarya Agrawal, Kate Saenko, Asli Celikyilmaz, and Vikas Chandra. 2024. An introduction to vision-language modeling. ArXiv, abs/2405.17247.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. 2024. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. arXiv preprint arXiv:2412.05271.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. 2024. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. arXiv preprint arXiv:2406.07476.

Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, et al. 2023. Mobilevlm: A fast, reproducible and strong vision language assistant for mobile devices. arXiv preprint arXiv:2312.16886.

Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. 2024a. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. arXiv preprint arXiv:2405.21075.

Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. 2024b. Framefusion: Combining similarity and importance for video token reduction on large visual language models. Preprint, arXiv:2501.01986.

Wenbo Hu, Zi-Yi Dou, Liunian Harold Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. 2024. Matryoshka query transformer for large vision-language models. arXiv preprint arXiv:2405.19315.

Xuan Ju, Yiming Gao, Zhaoyang Zhang, Ziyang Yuan, Xintao Wang, Ailing Zeng, Yu Xiong, Qiang Xu, and Ying Shan. 2025. Miradata: A large-scale video dataset with long durations and structured captions. Advances in Neural Information Processing Systems, 37:48955–48970.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2025. LLaVA-onevision: Easy visual task transfer. Transactions on Machine Learning Research.

Dongxu Li, Yudong Liu, Haoning Wu, Yue Wang, Zhiqi Shen, Bowen Qu, Xinyao Niu, Guoyin Wang, Bei Chen, and Junnan Li. 2024a. Aria: An open multimodal native mixture-of-experts model. arXiv preprint arXiv:2410.05993.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In International Conference on Machine Learning.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2023b. Videochat: Chat-centric video understanding. arXiv preprint arXiv:2305.06355.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. 2024b. Mvbench: A comprehensive multi-modal video understanding benchmark. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 22195–22206.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. 2024c. Token packer: Efficient visual projector for multimodal llm. arXiv preprint arXiv:2407.02392.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. Advances in neural information processing systems, 36:34892–34916.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. 2024a. Videogpt+: Integrating image and video encoders for enhanced video understanding. arXiv preprint arXiv:2406.09418.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2024b. Video-chatgpt: Towards detailed video understanding via large vision and language models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024).

OpenAI. 2023. Gpt-4v. https://openai.com/index/gpt-4v-system-card/.

OpenAI. 2024. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/.

Viorica Pătrăucean, Lucas Smaira, Ankush Gupta, Adrià Recasens Continente, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alex Frechette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, Yusuf Aytar, Simon Osindero, Dima Damen, Andrew Zisserman, and João Carreira. 2023. Perception test: A diagnostic benchmark for multimodal video models. In Advances in Neural Information Processing Systems.

Michael S Ryoo, Honglu Zhou, Shrikant Kendre, Can Qin, Le Xue, Manli Shu, Silvio Savarese, Ran Xu, Caiming Xiong, and Juan Carlos Niebles. 2024. xgen-mm-vid (blip-3-video): You only need 32 tokens to represent a video even in vlms. arXiv preprint arXiv:2410.16267.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2024. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. arXiv preprint arXiv:2403.15388.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. 2024. Dycoke: Dynamic compression of tokens for fast video large language models. ArXiv, abs/2411.15024.

Gemini Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. Preprint, arXiv:2403.05530.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. 2024. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. arXiv preprint arXiv:2406.16860.

Han Wang, Yuxiang Nie, Yongjie Ye, Deng GuanYu, Yanjie Wang, Shuai Li, Haiyang Yu, Jinghui Lu, and Can Huang. 2024a. Dynamic-vlm: Simple dynamic visual token compression for videollm. Preprint, arXiv:2412.09530.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024b. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. Preprint, arXiv:2409.12191.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. 2024c. Internvideo2: Scaling foundation models for multimodal video understanding. In European Conference on Computer Vision, pages 396–416. Springer.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. 2024. Longvlm: Efficient long video understanding via large language models. In European Conference on Computer Vision, pages 453–470. Springer.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. 2024. Longvideobench: A benchmark for long-context interleaved video-language understanding. arXiv preprint arXiv:2407.15754.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. Next-qa: Next phase of question-answering to explaining temporal actions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9777–9786.

Fuzhao Xue, Yukang Chen, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. 2024. Longvila: Scaling long-context visual language models for long videos. arXiv preprint arXiv:2408.10188.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024a. Qwen2.5 technical report. arXiv preprint arXiv:2412.15115.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2024b. Visionzip: Longer is better but not necessary in vision language models. arXiv preprint arXiv:2412.04467.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, Qianyu Chen, Huarong Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, Xu Han, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. Minicpm-v: A gpt-4v level mllm on your phone. Preprint, arXiv:2408.01800.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. Preprint, arXiv:2303.15343.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. 2024a. Lmms-eval: Reality check on the evaluation of large multimodal models. Preprint, arXiv:2407.12772.

Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. 2025. Llava-mini: Efficient image and video large multimodal models with one vision token. arXiv preprint arXiv:2501.03895.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. 2024b. Video instruction tuning with synthetic data. Preprint, arXiv:2410.02713.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. 2024c. Video instruction tuning with synthetic data. arXiv preprint arXiv:2410.02713.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. 2024. Mlvu: A comprehensive benchmark for multi-task long video understanding. arXiv preprint arXiv:2406.04264.

Kaitlyn Zhou, Kawin Ethayarajh, Dallas Card, and Dan Jurafsky. 2022. Problems with cosine as a measure of embedding similarity for high frequency words. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 401–423, Dublin, Ireland. Association for Computational Linguistics.